C++ bafta

1. #include <fstream>

```
using namespace std;
ifstream f("numere.in");
ofstream g("numere.out");
int n,i,v[100],aux;
void citire()
{f>>n;
 for(i=1;i<=n;i++)
 f>>v[i];
}
int pare(int n)
 { aux=n;
 while(aux)
 if(aux%10%2==0) aux=aux/10;
 else return 0;
 return 1;
 }
 void afis()
 { int maxi=0;
 for(i=1;i<=n;i++)
 {if (pare(v[i])==1) g<<v[i]<<" ";
 if(v[i]>maxi && pare(v[i])==1) maxi=v[i];}
 g<<endl<<maxi;
 }
int main()
{citire();
 afis();
 f.close(); g.close();
}
```

2. #include <fstream>

```
using namespace std;
ifstream f("numere.in");
ofstream g("numere.out");
int n,i,v[100],aux,nr;
void citire()
{f>>n;aux=n;
 }
 void afis()
 { aux/=10;
 while (aux)
 { g<<aux%10<<' ';
 aux/=100; }
 g<<endl;
 nr=n%10;
 n/=10;i=10;
 while(n>9)
 { nr=(n%10)*i+nr;
 n/=10;
 i=i*10;
 }
 nr=nr*10+n;
 g<<nr;
 }
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

3. #include <fstream>

```
using namespace std;
ifstream f("date.in");
ofstream g("date.out");
int n,i,v[100],mini=9999,k;
void citire()
{f>>n;
for(i=1;i<=n;i++)
 { f>>v[i];
 if(v[i]%2==1 && mini>v[i]) {mini=v[i]; k=i;}
 } }
 int cmmdc(int x,int y)
 {int r=x%y;
 while (r!=0)
 { x=y;
 y=r;
 r=x%y; }
 return y; }
 void afis()
 {if(k!=0) g<<mini<<' '<<k;
 else g<<"Nu exista";
 g<<endl<<cmmdc(v[1],v[2]); }
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

4. #include <fstream>

```
#include <cmath>
using namespace std;

ifstream f("date.txt");
ofstream g("date1.txt");

// Funcție pentru a calcula divizorii proprii ai unui număr
int divizori_proprii(int numar, int divizori[]) {
    int nr_divizori = 0;
    for (int i = 2; i <= sqrt(numar); ++i) {
        if (numar % i == 0) {
            divizori[nr_divizori++] = i;
            if (i != numar / i) { // Verificăm dacă i și numărul împărțit la i sunt diferite
                divizori[nr_divizori++] = numar / i;
            }
        }
    }
    return nr_divizori;
}

int main() {
    // Citirea datelor din fișier
    int numar;
    int max_divizori[50]; // Vector pentru a stoca numerele cu cele mai multe divizori proprii
    int max_nr_divizori = -1;
    int nr_max_numere = 0;

    while (f >> numar) {
```

```
        g << numar << " ";

        int divizori[50]; // Vector pentru a stoca divizorii proprii ai numărului curent
        int nr_divizori = divizori_proprii(numar, divizori);

        for (int i = 0; i < nr_divizori; ++i) {
            g << divizori[i] << " ";
        }
        g << endl;

        // Actualizăm numărul maxim de divizori și vectorul cu numerele cu cele mai
multe divizori
        if (nr_divizori > max_nr_divizori) {
            max_nr_divizori = nr_divizori;
            nr_max_numere = 0;
            max_divizori[nr_max_numere++] = numar;
        } else if (nr_divizori == max_nr_divizori) {
            max_divizori[nr_max_numere++] = numar;
        }
    }

    // Afișare numărul cu cei mai mulți divizori proprii, dar cel mai mic în caz de
egalitate
    int min_numar = max_divizori[0];
    for (int i = 1; i < nr_max_numere; ++i) {
        if (max_divizori[i] < min_numar) {
            min_numar = max_divizori[i];
        }
    }
    g << min_numar;
```

```
    f.close();
    g.close();
    return 0;
}
```

4 care e tot 4.

```
#include <fstream>
using namespace std;

ifstream f("numere.txt");
ofstream g("Rezultate.txt");

void citire(int v[], int& n) {
    n = 0;
    while (f >> v[n]) {
        ++n;
    }
}

bool prim(int n) {
    if (n < 2) {
        return false;
    }
    for (int i = 2; i * i <= n; ++i) {
        if (n % i == 0) {
            return false;
        }
    }
    return true;
}
```

```
void afisare_primul_prim(int v[], int n) {
    bool gasit = false;
    for (int i = 0; i < n; ++i) {
        if (prim(v[i])) {
            g << v[i] << endl;
            gasit = true;
            break;
        }
    }
    if (!gasit) {
        g << "NU EXISTA" << endl;
    }
}

void afisare_descompunere(int n) {
    for (int i = 2; i <= n; ++i) {
        int exp = 0;
        while (n % i == 0) {
            n /= i;
            ++exp;
        }
        if (exp > 0) {
            g << i << " " << exp << endl;
        }
    }
}

int main() {
    int v[1000];
```

```
    int n;

    citire(v, n);
    afisare_primul_prim(v, n);

    if (n > 0) {
        afisare_descompunere(v[0]);
    }

    f.close();
    g.close();
    return 0;
}
```

6.
```
#include <fstream>
using namespace std;
ifstream f("numere.txt");
ofstream g("numere1.txt");
int n,i,v[100],aux,c, maxi,j;
void citire()
{while(f>>n)
 v[++i]=n;
}
int palindrom(int n)
{int inv=0,aux=n;
 while(aux)
 {
 inv=inv*10+aux%10;
 aux=aux/10;
 }
```

```
 if(inv==n) return 1;
 return 0;
}
 void afis()
 {
 for(j=1;j<=i;j++)
 if (v[j]>maxi && palindrom(v[j])==1) maxi=v[j];
 g<<maxi<<endl;
 for(j=1;j<=i;j++) if(palindrom(v[j])==1) {g<<v[j]<<" ";
 break;}
 for(j=i;j>=1;j--) if(palindrom(v[j])==1){g<<v[j]; break;}
}
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

```
7. #include <fstream>
using namespace std;
ifstream f("atestat.txt");
ofstream g("atestat1.txt");
int n,i,v[100],k;
void citire()
{f>>n;
 for(i=1;i<=n;i++)
 f>>v[i];
}
int nrcif(int n)
```

```
{int nr=0,aux=n;
 while (aux)
 { nr++;
 aux=aux/10;
 }
 return nr;
 }
 int cmmdc(int x,int y)
 {int r=x%y;
 while (r!=0)
 { x=y;
 y=r;
 r=x%y; }
 return y; }
 void afis()
 { int maxi=0, mini=9999;
 for(i=1;i<=n;i++)
 {if(v[i]%2!=0) g<<nrcif(v[i])<<" ";
 if(v[i]>maxi)maxi=v[i];
 if(v[i]<mini) mini=v[i];
 }
g<<endl;
g<<cmmdc(mini,maxi);
}
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

```
8. #include <fstream>
using namespace std;
ifstream f("cmmdc3.txt");
ofstream g("cmmdc31.txt");
int n,a,b,c;
int cmmdc(int x,int y)
{ int r=x%y;
 while (r)
 { x=y;
 y=r;
 r=x%y;
 }
 return y;
}
void citire()
{ f>>n;
 a=n%1000;
 n=n/1000;
 b=n%1000;
 c=n/1000;
}
void afis()
{ g<<c<<' '<<b<<' '<<a<<endl;
g<<cmmdc(cmmdc(a,b),c);
}
int main()
{citire();
 afis();
 f.close();
```

```
 g.close();
}
```

9. #include <fstream>

```
using namespace std;
ifstream f("tip1.txt");
ofstream g("tip11.txt");
int n,k,nr,i=1,aux;
void citire()
{ f>>n>>k;
 aux=n;
}
void afis()
{ while(aux)
{ if(aux%10!=k) {nr=aux%10*i+nr;
 i=i*10;}
aux/=10;
}
g<<nr<<endl;
i=k;
while(i<n) i=k*i;
i=i/k;
g<<i;
}
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

10. #include <fstream>

```
using namespace std;
ifstream f("numere.txt");
ofstream g("numere1.txt");
int n,i,v[100];
void citire()
{f>>n;
 for(i=1;i<=n;i++)
 f>>v[i];
}
int pal(int a)
{int aux=a; int inv=0;
while (aux)
{ inv=inv*10+aux%10;
 aux/=10;}
 if(inv==a) return 1;
 return 0;
}
int cifr(int x)
{ int k=0;
 while (x)
 {k++;
 x/=10;
 }
 return k;
}
void afis()
{ for(i=1;i<=n;i++)
if(pal(v[i])==1) g<<v[i]<<' ';
```

```
g<<endl;
for(i=1;i<n;i++)
 for(int j=i+1;j<=n;j++)
if(v[j]<v[i]) swap(v[i],v[j]);
for(i=1;i<=n;i++)
 if(pal(v[i])==1 && cifr(v[i])%2==1) g<<v[i]<<' ';
}
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

```
11. #include <fstream>
#include <cmath>
using namespace std;

ifstream f("puncte.txt");
ofstream g("puncte1.txt");

struct coord {
    int x, y;
} v[100];

int i = 1;

void citire() {
    int x, y;
    while (f >> x >> y) {
```

```
        v[i].x = x;
        v[i].y = y;
        i++;
    }
}

int distante(coord v[], int index1, int index2) {
    int d = sqrt(pow((v[index1].x - v[index2].x), 2) + pow((v[index1].y - v[index2].y),
2));
    return d;
}

void afis() {
    int q = 0, maxi = 0;
    for (int k = 1; k < i; k++) {
        if (v[k].x >= 0 && v[k].y >= 0) {
            q++;
        }
    }
    g << q;
    for (int k = 1; k < i; k++) {
        for (int j = k + 1; j < i; j++) {
            if (maxi < distante(v, k, j)) {
                maxi = distante(v, k, j);
            }
        }
    }
    g << " " << maxi;
}
```

```
int main() {
    citire();
    afis();
    f.close();
    g.close();
    return 0;
}
```

12. #include <fstream>

```
#include <iostream>
using namespace std;

ifstream f("fractii.txt");
ofstream g("fractii1.txt");

int n, i;

struct fractie {
    int a, b;
} v[100];

void citire() {
    f >> n;
    for (i = 1; i <= n; i++) {
        f >> v[i].a >> v[i].b;
    }
}

int cmmdc(int x, int y) {
    int r = x % y;
```

```
    while (r) {
        x = y;
        y = r;
        r = x % y;
    }
    return y;
}

int cmmmc(int x, int y) {
    return x * y / cmmdc(x, y);
}

void afis() {
    int k = 0;
    for (i = 1; i <= n; i++) {
        if (cmmdc(v[i].a, v[i].b) == 1) {
            k++;
        }
    }
    g << k << ' ';
    k = cmmmc(v[1].b, v[2].b);
    for (i = 3; i <= n; i++) {
        k = cmmmc(k, v[i].b);
    }
    g << k;
}

int main() {
    citire();
    afis();
```

```
    f.close();
    g.close();
    return 0;
}
```

```
13. #include <fstream>
using namespace std;
ifstream f("atestat.txt");
ofstream g("atestat1.txt");
struct coordonate{int x,y,r;}v[100];
int i,j,n;
void citire()
{f>>n;
for(int i=1;i<=n;i++)
 f>>v[i].x>>v[i].y>>v[i].r;
}
void ordonare()
{for(i=1;i<=n;i++)
 for(j=i+1;j<=n;j++)
 if(v[i].r>v[j].r) swap(v[i],v[j]);
 for(i=1;i<=n;i++)
 g<<"("<<v[i].x<<","<<v[i].y<<","<<v[i].r<<")" << endl;
}
void afis()
{int k=0;
for(i=1;i<=n;i++)
 if(v[i].x==v[i].y) k++;
g<< k;
}
int main()
```

```
{citire();
 ordonare();
 afis();
 f.close();
 g.close();
}
```

14. #include <iostream>

```
#include<fstream>
using namespace std;
ifstream f("tip5.txt");
ofstream g("tip51.txt");
int n,k,ok;
struct prod
{int cod,pret,cant;
char den[30];
}v[100],aux;
void citire()
{f>>n;
for(int i=1;i<=n;i++)
{f>>v[i].cod>>v[i].pret>>v[i].cant>>v[i].den;
k+=v[i].cant*v[i].pret;
}}
void ord()
{for(int i=1;i<n;i++)
for(int j=i+1;j<=n;j++)
if(v[i].pret<v[j].pret)swap(v[i],v[j]);
for(int i=1;i<=n;i++)cout<<v[i].cod<<' '<<v[i].den<<' '<<v[i].pret<<'
'<<v[i].cant<<endl;
}
```

```
int main()
{citire();
for(int i=1;i<=n;i++)if(v[i].cant>10){cout<<v[i].den<<", ";ok=1;}
if(ok==0)cout<<"NU EXISTA";
cout<<endl;
ord();
cout<<k;
return 0;
}
```

15.
```
#include <fstream>
#include <cstring>
#include <iomanip>
using namespace std;
ifstream f("elev.in");
ofstream g("elev.out");
int n,i,k;
struct elev{char nume[40],prenume[25]; float medie;};elev v[30];
void citire()
{f>>n;
 for(i=1;i<=n;i++)
 {int s=0;
 for(int j=1;j<=5;j++)
{f>>k; s=s+k;
}
v[i].medie=s/5;
f>>v[i].nume;
f.get();
f>>v[i].prenume;
}}
```

```
void ord1(elev v[30], int n)
{ for(i=1;i<n;i++)
for(int j=i+1;j<=n;j++)
 if(v[i].medie<v[j].medie) swap(v[i],v[j]);
}
void ord2(elev v[30],int n)
{for(i=1;i<n;i++)
for(int j=i+1;j<=n;j++)
 if(strcmp(v[i].nume,v[j].nume)>0) swap(v[i],v[j]);
}
void afis()
{ ord1(v,n);
 for(i=1;i<=n;i++)
 g<<fixed<<setprecision(2)<<v[i].medie<<' '<<v[i].nume<<'
'<<v[i].prenume<<endl;
 ord2(v,n);
 for(i=1;i<=n;i++)
 g<<fixed<<setprecision(2)<<v[i].medie<<' '<<v[i].nume<<'
'<<v[i].prenume<<endl;
}
int main()
{citire();
 afis();
 f.close();
 g.close();
}
```

16.
```
#include <fstream>
using namespace std;
```

```
ifstream f("numere.txt");
ofstream g("numere1.txt");

int n, v[100], a[100], maxi = -1; // Inițializează variabila maxi cu o valoare negativă

void citire() {
    f >> n;
    for (int i = 1; i <= n; i++) {
        f >> v[i];
    }
}

int suma(int x) {
    int s = 0;
    while (x) {
        s += x % 10;
        x /= 10;
    }
    return s;
}

void afis() {
    for (int i = 1; i <= n; i++) {
        g << suma(v[i]) << " ";
        a[i] = suma(v[i]);
        if (a[i] > maxi) { // Actualizează valoarea maximă a sumei
            maxi = a[i];
        }
    }
    g << endl;
```

```
    for (int i = 1; i <= n; i++) {
        if (a[i] == maxi) {
            g << v[i] << " ";
        }
    }
}

int main() {
    citire();
    afis();
    f.close();
    g.close();
    return 0;
}
```

17. #include <iostream>

```
#include <fstream>
using namespace std;
ifstream f("atestat.txt");
ofstream g("atestat1.txt");
int n,v[100],i,j,a[100];
void citire()
{ f>>n;
 for(i=1;i<=n;i++)
 f>>v[i];
}
void afis()
{ int s=0;
 for(i=1;i<=n;i=i+2)
{s=s+v[i]; j++; s=s/j;}
```

```
cout<<s<<endl;
for(i=1;i<n;i++)
 for(j=i+1;j<=n;j++)
 if(v[i]>v[j]) swap (v[i],v[j]);
 cout<<v[1]<<' ';
for(i=2;i<=n;i++)
 if(v[i]!=v[i-1]) cout<<v[i]<<' ';
}
int main()
{ citire();
afis();
}
```

18. #include <iostream>

```
#include <fstream>
using namespace std;
ifstream f("numere.in");
ofstream g("numere.out");
int n, v[100],i,j,ok;
int prime(int a,int b)
{int r=a%b;
while(r!=0)
{a=b;
b=r;
r=a%b;
}
return b;
}
void citire()
{f>>n;
```

```
for(i=1;i<=n;i++)
f>>v[i];
}
void solve(int v[100], int &n)
{for(i=1;i<n;i++)
if(prime(v[i],v[i+1])!=1){ok=1;
for(j=i;j<n;j++)
v[j]=v[j+1];
n--;}
}
int main()
{citire();
solve(v,n);
if(ok==1){g<<"NU"<<endl;
for(i=1;i<=n;i++)
g<<v[i]<<' ';
}
else g<<"DA";
}
```

19.
```
#include <iostream>
#include <fstream>
using namespace std;
ifstream f("atestat.txt");
ofstream g("atestat1.txt");
int n,v[100],i,j,a[100];
void citire()
{f>>n;
 for(i=1;i<=n;i++)
```

```
 f>>v[i];
}
int cifr(int x)
{int k=0;
while(x)
{ k++;
 x/=10;
}
 return k;
}
void afis()
{for(i=n;i>=1;i--)
 g<<v[i]<<' ';
g<<endl;
 for(i=1;i<=n;i++)
 g<<cifr(v[i])<<' ';
}
int main()
{ citire();
afis();
}
```

20. #include <iostream>

```
#include <fstream>
using namespace std;
ifstream f("numere.in");
int n=1,v[101],k,aux,ok;
void citire()
{while(f>>v[n])n++;
n--;
```

```
}
int div(int x)
{int ok=0;
int aux=x;
while(aux)
{ if (aux%10==0)ok=1;
aux/=10;
}
return ok;
}
void afis()
{for(int i=1;i<=n;i++)
if(div(v[i]))k++;
cout<<k<<endl;
for(int i=1;i<=n;i++)
{aux=v[i];
while(aux)
{if(aux==v[i+1])ok=1;
aux/=10;
}
if(ok==1){cout<<v[i]<<' '<<v[i+1];break;}
}
}
int main()
{citire();
afis();
return 0;
}
```

21. #include <fstream>

```
#include <cmath>
using namespace std;
ifstream f("date.in");
ofstream g("date.out");
int n,v[100],i,j,a[100];
void citire()
{f>>n;
 for(i=1;i<=n;i++)
 f>>v[i];
}
int cifr(int x)
{int k=1;
while(x)
{k=k*10;
 x/=10;
}
 return k/10;
}
int prima_cifra(int x)
{while (x>9){x/=10;}
 return x;
}
void afis()
{ int nr=0;
 for(i=1;i<=n;i++)
 g<<v[i]%cifr(v[i])<<' ';
g<<endl;
 for(i=1;i<=n;i++)
 a[prima_cifra(v[i])]++;
```

```
 for(i=9;i>=0;i--)
{if(a[i]!=0) nr=nr*10+i;
}
g<<nr;
}
int main()
{ citire();
afis();
}
```

22.
```
#include <fstream>
#include <iostream>
#include <cmath>
using namespace std;

ifstream f("numere.txt");
ofstream g("numere1.txt");

int n, k;

// Funcție pentru a verifica dacă un număr este prim
bool prim(int x) {
    if (x < 2) return false;
    for (int i = 2; i <= sqrt(x); ++i) {
        if (x % i == 0) return false;
    }
    return true;
}
```

```
// Funcție pentru a verifica dacă cifrele unui număr au aceeași paritate
bool aceeasi_paritate(int x) {
    int cifra;
    bool paritate = x % 2 == 0; // Paritatea primului cifre
    x /= 10; // Trecem la următoarea cifră
    while (x > 0) {
        cifra = x % 10;
        if ((cifra % 2 == 0) != paritate) { // Verificăm dacă paritatea difera de la prima
cifră
            return false;
        }
        x /= 10;
    }
    return true;
}

// Funcție pentru a determina numărul de divizori proprii ai unui număr
int numar_divizori(int x) {
    int divizori = 0;
    for (int i = 1; i <= sqrt(x); ++i) {
        if (x % i == 0) {
            if (i != x / i) {
                divizori += 2;
            } else {
                divizori++;
            }
        }
    }
    return divizori;
}
```

```
int main() {
    f >> n >> k;
    int max_divizori = 0;
    int nr;
    f >> nr; // Citim primul număr

    // Verificăm dacă cifrele primului număr au aceeași paritate
    if (aceeasi_paritate(nr)) {
        cout << "DA\n";
    } else {
        cout << "NU\n";
    }

    // Scriem primul număr în fișierul de ieșire dacă are cel puțin k divizori
    if (numar_divizori(nr) >= k) {
        g << nr << " ";
        max_divizori = numar_divizori(nr);
    }

    // Citim celelalte numere și verificăm numărul de divizori
    for (int i = 1; i < n; ++i) {
        f >> nr;
        int div = numar_divizori(nr);
        if (div >= k) {
            if (div > max_divizori) {
                max_divizori = div;
                g << nr << " ";
            } else if (div == max_divizori) {
                bool exista = false;
```

```
            // Verificăm dacă numărul există deja în listă
            f.seekg(0, ios::beg);
            int numar;
            while (f >> numar) {
                if (numar == nr) {
                    exista = true;
                    break;
                }
            }
            if (!exista) {
                g << nr << " ";
            }
        }
    }
}

f.close();
g.close();

return 0;
}
```

23. #include <fstream>

```
using namespace std;
ifstream f("n.txt");
ofstream g("n1.txt");
int n, v[100],a[100],k,z;
void citire()
{ f>>n;
for(int i=1;i<=n;i++)
```

```
 f>>v[i];
}
void paritate()
{for(int i=1;i<=n;i++)
if(v[i]%2==0)a[++z]=v[i];
for(int i=1;i<z;i++)
for(int j=i;j<=z;j++)
if(a[i]<a[j])swap(a[i],a[j]);
for(int i=1;i<=z;i++)
 g<<a[i]<<" ";
 g<<endl;
}
void supraperfect()
{for(int i=1;i<=n;i++)
{int suma=0;
for(int d=1;d<=v[i]/2;d++)
if(v[i]%d==0)suma=suma+d;
if(suma>v[i]){k=1;g<<v[i]<<" ";}
}
if(k==0)g<<"NU EXISTA NUMERE SUPRAPERFECTE";
}
int main()
{ citire();
paritate();
supraperfect();
}
```

24.
```
#include <fstream>
using namespace std;
ifstream f("numere.txt");
```

```
ofstream g("numere1.txt");
int n, v[100],a[100],k;
void citire()
{f>>n;
for(int i=1;i<=n;i++)
 f>>v[i];
}
void verificare()
{ for(int i=1;i<n;i++)
for(int j=i;j<=n;j++)
if(v[i]>v[j]){k=1;break;}
if(k==0)g<<"Da";
else g<<"Nu"; g<<endl;
}
int nr_cif(int x)
{int aux=x,k=0;
while(aux)
 {k++; aux/=10; }
return k;
}
void afisare()
{for(int i=1;i<=n;i++)
 {int c=nr_cif(v[i])-1,p=1;
while(c)
 {p=p*10;
 c--; }
v[i]=v[i]/p; }
for(int i=1;i<n;i++)
for(int j=i;j<=n;j++)
if(v[i]<v[j]) swap(v[i],v[j]);
```

```
for(int i=1;i<=n;i++)
 g<<v[i];
}
int main()
{citire();
verificare();
afisare();
}
```

25. #include <fstream>

```
using namespace std;
ifstream f("n.in");
ofstream g("n.out");
long long n,aux;
int suma,cifmax=0,fr[10];
void sumcif()
{while(aux)
{ if(aux%10>cifmax)cifmax=aux%10;
suma=suma+aux%10;
aux/=10; }
g<<suma<<" "<<cifmax<<endl;;
}
voidfrecventa()
{while(n)
{int c=n%10;
fr[c]++;
n/=10; }
for(int i=9;i>=1;i--)
 g<<fr[i]<<" ";
}
```

```
int main()
{ f>>n;
aux=n;
sumcif();
frecventa();
}
```